



Machine Learning Course By Dr. Nazerfard
CE5501 | Fall 2023
Teaching Assistants
AMIRHOESSION BABYIAN

دانشجو: نیما پری فرد

شماره دانشجویی: ۴۰۲۱۳۱۰۱۷

ایمیل: nima.parifard@aut.ac.ir

فهرست ارائه نهایی پروژه درس یادگیری ماشین

- بخش ۱ بیان چالش های پیش آمده:..... ۲
- بخش ۲ شبیه سازی و ترسیم گراف مترو:..... ۳
- بخش ۳ ساخت دیتاست:..... ۶
- بخش ۴ تربیت مدل یادگیری ماشین برای خروجی هر ایستگاه:..... ۷
- بخش ۵ تربیت مدل یادگیری ماشین برای ورودی هر ایستگاه:..... ۷
- بخش ۶ تربیت مدل یادگیری ماشین برای افراد حاضر در مترو:..... ۸
- بخش ۷ تربیت مدل یادگیری ماشین برای تخمین نرخ پواسون:..... ۸
- بخش ۸ ارزیابی:..... ۹

لینک گیت هاب پروژه: <https://github.com/nimaiparifard/cursed-metro-ml-final-project>

بخش ۱ بیان چالش های پیش آمده:

۱- اینکه در تایم های اول صبح که هنوز اولین قطار ها به ایستگاه ها نرسیده خروجی از آن ایستگاه نداریم.

راه حل: من با استفاده یک دیتاست که در آن مشخص کردم هر قطار چه ساعتی به ایستگاه می رسد مشکل خود به خود می شود. و واضح چون در آن ساعت کسی وارد نشده

```
train_number,line_number,current_timestamp,current_station,current_passenger,next_station,next_station_time_arrived
1_1,1,2024-01-01 06:00:00,1,31,2,2024-01-01 06:06:00
1_1,1,2024-01-01 06:06:00,2,34,3,2024-01-01 06:12:00
1_1,1,2024-01-01 06:12:00,3,29,4,2024-01-01 06:18:00
1_1,1,2024-01-01 06:18:00,4,28,5,2024-01-01 06:24:00
```

۲- در ایستگاه های مشترک خروجی ها چگونه تقسیم شوند

راه حل: در کلاس بحث شد که به دوقسمت تقسیم می شوند من با استفاده از Children گزاشتن روی ایستگاه ها خروجی را تقسیم بر تعداد بچه های هر ایستگاه می کردم.

```
self.passengers -= self.output_rate
current_passenger = int((current_passenger + self.passengers) / len(self.children))
if len(self.children) > 1:
    current_passenger = int((current_passenger + self.passengers) / len(self.children))
```

۳- نرخ خروجی هر قطار در یک ایستگاه نباید بیش تر از افراد داخل آن قطار شوند
راه حل: با استفاده از همان فایل CSV من در لحظه افراد حاضر در هر قطار را داشتم و یک شرط گزاشتم که خروجی از افراد حاضر در قطار بیش تر نشود.

```
output_rate = int(np.random.poisson(np.random.randint(self.output_rate_range[0],
self.weekend_rate * self.holiday_rate * self.crowded_station_rate)))
if output_rate > current_passengers:
    output_rate = current_passengers
```

۴- وقتی قطار به ایستگاه آخر می رسد باید همه افرادی که داخل هستند خارج شوند

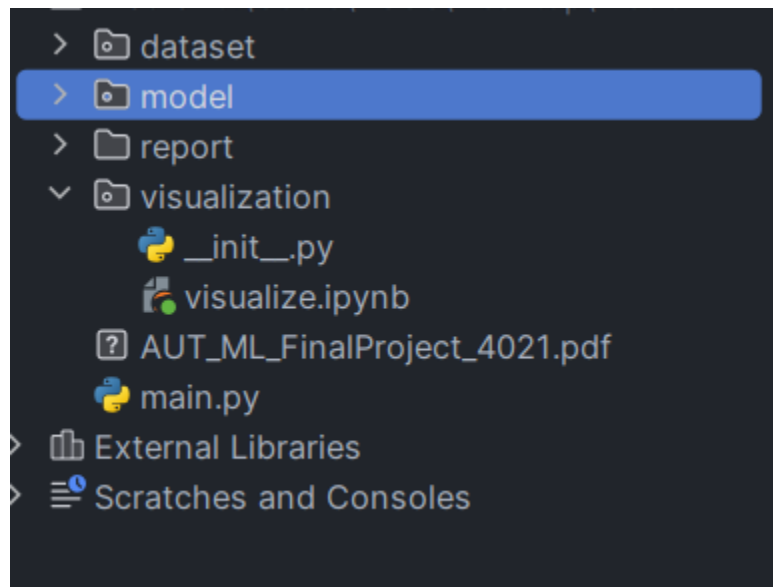
```
elif self.is_last_station:
    output_rate = current_passengers
```

وقتی نرخ خروجی را می خواستم تعیین کنم اول چک می کردم بینم در ایستگاه آخر هست یا خیر
۵- در تایم آخر بعد از اینکه در بعضی ایستگاه ها آخرین قطار رد شد دیگر ورودی در آن ایستگاه ها نداریم و مترو بسته می شود
با استفاده از همان dataset می توان اینکار به طور اتوماتیک حل می شود.

بخش ۲ شبیه سازی و ترسیم گراف مترو:

شبیه سازی با حل مشکلات قبل به دست آمد اگر فایل metro.py اجر کنید شبیه سازی نشان داده می شود.
پروژه بنده دارای سه ماژول dataset که شبیه سازی در آن انجام شده است. دارای سه فایل مهم است.
Station.py که عملیات هایی که برای هر ایستگاه نیاز است انجام می دهیم. در metro.py جریان شبیه سازی انجام می شود. در نهایت یک فایل CSV به ما می دهد که داده های شبیه سازی در آن بارگزاری شده است. و یک train_status_creation.py که فایل حرکت قطار و زمان رسیدنشون با ایستگاه های متخلف در یک روز بارگزاری می کند.

در ماژول مدل ۴ فایل جوینتر وجود دارد که هر کدام مدل های ورودی و خروجی و پواسون و افراد حاضر در ایستگاه را پیش بینی می کند.



در کلاس Station سه تابع مهم وجود دارد یکی برای تولید نرخ ورودی و تولید نرخ خروجی و همین طور تابعی برای اینکه جریان مسافر را تولید کند.

```
1 usage (1 dynamic)  Nima Parifard
def generate_input_rate(self):_# these people came from outsied of the metro rate ...
    input_rate = ____int(np.random.poisson(np.random.randint(self.input_rate_range[0], self.input_rate_range[1]+1)) * se
                        self.weekend_rate * self.holiday_rate * self.crowded_station_rate)
    if self.is_last_station:
        input_rate = 0
    logging.info(f'the input (from out of the metro) into the station {self.name} is : {input_rate}')
    self.input_rate = input_rate
```

```
1 usage (1 dynamic)  Nima Parifard
def generate_output_rate(self, current_passengers):_# these people leave the metro rate ....
    output_rate = 0
    if self.is_first_station:
        output_rate = 0
    elif self.is_last_station:
        output_rate = current_passengers
        logging.info(f'END OF THE LINE: the output (leaving the metro) from the station {self.name} is : {output_rate}')
    else:
        output_rate = int(np.random.poisson(np.random.randint(self.output_rate_range[0], self.output_rate_range[1]+1)) * se
                        self.weekend_rate * self.holiday_rate * self.crowded_station_rate)
        if output_rate > current_passengers:
            output_rate = current_passengers
        logging.info(f'the output (leaving the metro) from the station {self.name} is : {output_rate}')
    self.output_rate = output_rate
```

```

def passengers_flow(self, timestamp, train_number, current_passenger):
    change_line_passengers = self.passengers
    print(f'passenger flow started for {self.name}')
    self.passengers += self.input_rate

    # step 2: the passengers which leaves the metro through this station ...
    self.passengers -= self.output_rate
    current_passenger = int((current_passenger + self.passengers) / len(self.children))
    if len(self.children) > 1:
        current_passenger = int((current_passenger + self.passengers) / len(self.children))
    if self.is_last_station:
        current_passenger = 0
    format = '%Y-%m-%d %H:%M:%S'

    df = pd.read_csv("train_status.csv")
    df.loc[(df['train_number'] == train_number) & (df['current_timestamp'] == timestamp.strftime(format)), 'current_passenger'] = current_passenger
    df.to_csv("train_status.csv", index=False)

    # Open or create the CSV file in append mode
    with open('metro_passenger_flow.csv', 'a', newline='') as file:
        writer = csv.writer(file)
        # Check if file is empty to write headers
        if file.tell() == 0:
            writer.writerow(['timestamp', 'station_name', 'input_count', 'output_count', 'line_number', 'crowded_time_rate'])
        # Write the data row
        writer.writerow([timestamp, self.name, self.input_rate, self.output_rate, self.line_number, self.crowded_rate, self])

```

در کلاس Metro اول از همه تمام ایستگاه با استفاده از کلاس station تمام ایستگاه ها اول مقداردهی می شوند. و بعد با تابع metro_simulation_one_day شبیه سازی انجام می دهیم.

```

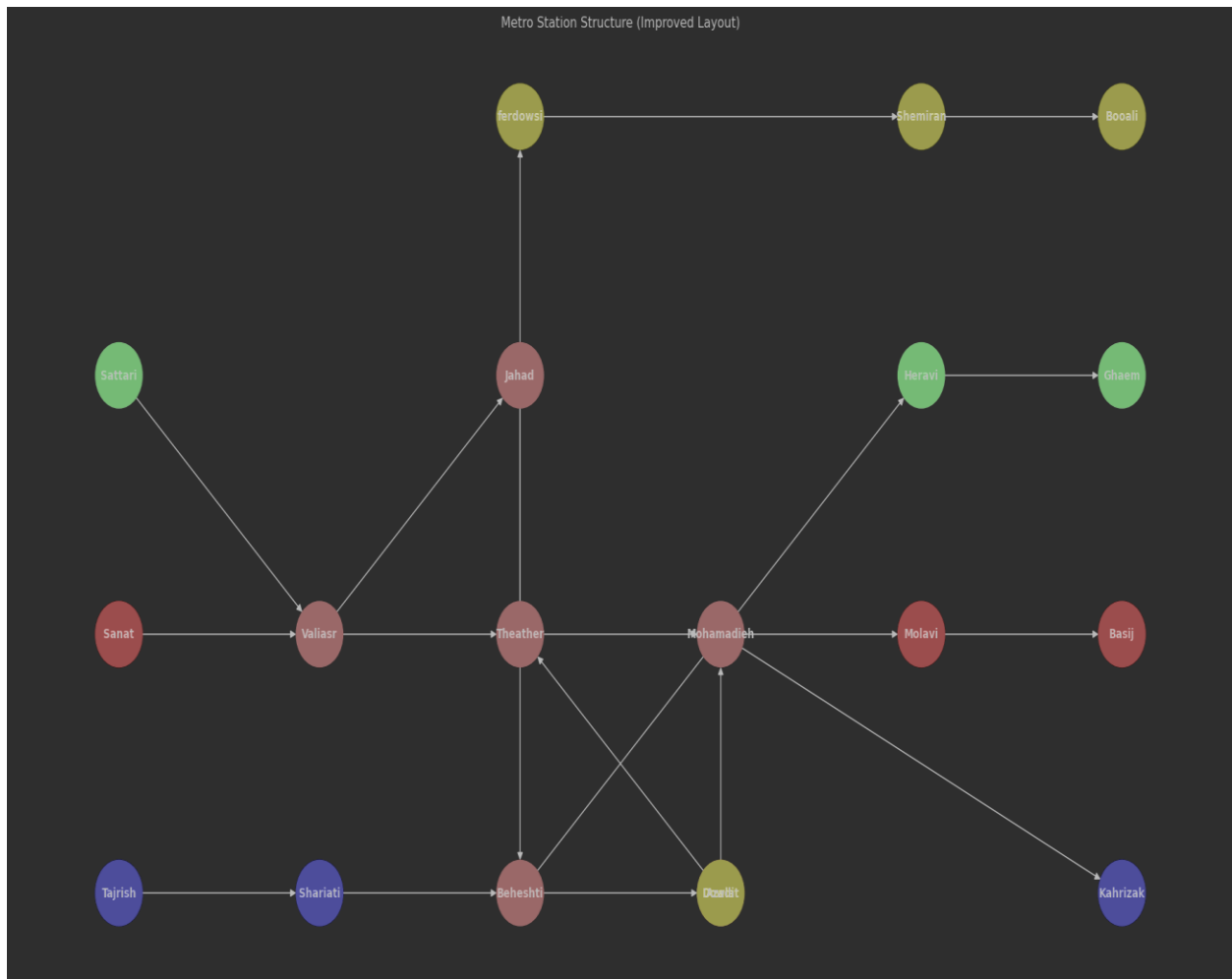
def metro_simulation_one_day(self):
    # Start time: 6 AM on 1st Jan 2024
    start_time = datetime.datetime(2024, 1, 1, 6, 0)
    # End time: 10 PM on the same day
    end_time = datetime.datetime(2024, 1, 1, 21, 30)

    # Current simulation time
    timestamp = start_time
    train_number = 0
    while timestamp <= end_time:
        print(f"Simulating for Holiday time: {timestamp}") # Placeholder for simulation steps

        # Loop through each line in sequence
        train_number += 1
        for line_name, stations in self.lines.items(): # Assuming self.lines is defined in __init__
            print(f"Processing {line_name}") # Placeholder for line processing
            current_passenger = 0
            line_timestamp = timestamp
            for station_name in stations:
                station_number_in_line, line_number = self.get_station_number_in_line(station_name, line_name)
                train_number_ = str(line_number) + "_" + str(train_number)
                station = self.stations[station_name]
                current_passenger = self.simulate_station_process(station, line_name, line_timestamp, train_number_, current_passenger)
                line_timestamp += datetime.timedelta(minutes=6)
            print(f"Simulated passengers flow at {station.name}") # Placeholder for actual simulation

```

گراف پیاده سازی مترو:



بخش ۳ ساخت دیتاست:

برای ساخت دیتاست من خودم ویژگی های دیگری اضافه کردم مثل اینکه آیا در ساعت شلوغی هست یا خیر. روز تعطیل است یا خیر و روز آخر هفته است یا خیر شماره خط ایستگاه و اینکه آیا جزو ایستگاه های شلوغ هست یا خیر. ایستگاه های شلوغ بعضی ایستگاه ها هستند که خودم مشخص کردم.

```
timestamp,station_name,input_count,output_count,line_number,crowded_time_rate,is_crowded_station,is_weekend,is_holiday
2024-01-01 06:00:00,Tajrish,31,0,Line1,0.8,True,0,0
2024-01-01 06:06:00,Shariati,9,6,Line1,0.8,False,0,0
2024-01-01 06:12:00,Beheshti,5,10,Line1,0.8,False,0,0
2024-01-01 06:18:00,Dowlat,11,12,Line1,0.8,True,0,0
2024-01-01 06:24:00,Mohamadieh,18,17,Line1,0.8,True,0,0
2024-01-01 06:30:00,Kahrizak,0,29,Line1,0.8,False,0,0
2024-01-01 06:00:00,Sanat,39,0,Line2,0.8,False,0,0
2024-01-01 06:06:00,Valiasr,11,14,Line2,0.8,True,0,0
2024-01-01 06:12:00,Theather,10,8,Line2,0.8,True,0,0
2024-01-01 06:18:00,Mohamadieh,13,14,Line2,0.8,True,0,0
```

بخش ۴ تربیت مدل یادگیری ماشین برای خروجی هر ایستگاه:

اول از همه به علت کم بودن ویژگی ها از یک کرنل دو بعدی برای بالا بردن تعداد ویژگی ها و بعد آن ها استفاده کردم. بعد سه تا مدل رگرسیون خطی و xgboost و random forest استفاده کردم و نتایج زیر به دست آمد.

```
Executed at 2024.02.14 16:23:24 in 8ms

XGBoost RMSE: 4.204138694193032
XGBoost R-squared: 0.8748518313558644
Linear Regression RMSE: 4.184607250624892
Linear Regression R-squared: 0.8760119484174724
Random Forest Regression RMSE: 4.529859968933572
Random Forest Regression R-squared: 0.8547085751907065
```

بخش ۵ تربیت مدل یادگیری ماشین برای ورودی هر ایستگاه:

اول از همه به علت کم بودن ویژگی ها از یک کرنل دو بعدی برای بالا بردن تعداد ویژگی ها و بعد آن ها استفاده کردم. بعد سه تا مدل رگرسیون خطی و xgboost و random forest استفاده کردم و نتایج زیر به دست آمد.

```
XGBoost RMSE: 6.476383100070873
XGBoost R-squared: 0.631714471457155
Linear Regression RMSE: 6.419743543864982
Linear Regression R-squared: 0.6381280249816235
Random Forest Regression RMSE: 6.601959386761759
Random Forest Regression R-squared: 0.6172939841954003
```

بخش ۶ تربیت مدل یادگیری ماشین برای افراد حاضر در مترو:

اول از همه باید دیتاست آماده می کردم که در هر لحظه چند نفر داخل ایستگاه هستند. بعد برای بقیه مدل ها مانند دو روش قبلی عمل کردم.

```
XGBoost RMSE: 51.65299131592512
XGBoost R-squared: 0.8486811157062791
Linear Regression RMSE: 53.612635848832205
Linear Regression R-squared: 0.8369816492029062
Random Forest Regression RMSE: 48.36787373133901
Random Forest Regression R-squared: 0.8673167294790962
```

بخش ۷ تربیت مدل یادگیری ماشین برای تخمین نرخ پواسون:

برای نرخ پواسون از مدل های یادگرفته شده استفاده کردم میانگین در دیتاست برای هر ایستگاه محاسبه کردم و به عنوان ورودی مدل دادم و در نهایت خروجی را دریافت کردم. به عنوان نرخ پواسون معرفی شده است.

بخش ۸ ارزیابی:

برای ارزیابی از دو معیار $rmse$ و r^2 استفاده کردم که در شکل برای هر مدل این معیاری ها را مشاهده می کنید. معیار r^2 به خاطر اینکه بین صفر و یک قرار می گیرد حس بهتری به ما می دهد نسبت به سایر مدل ها.